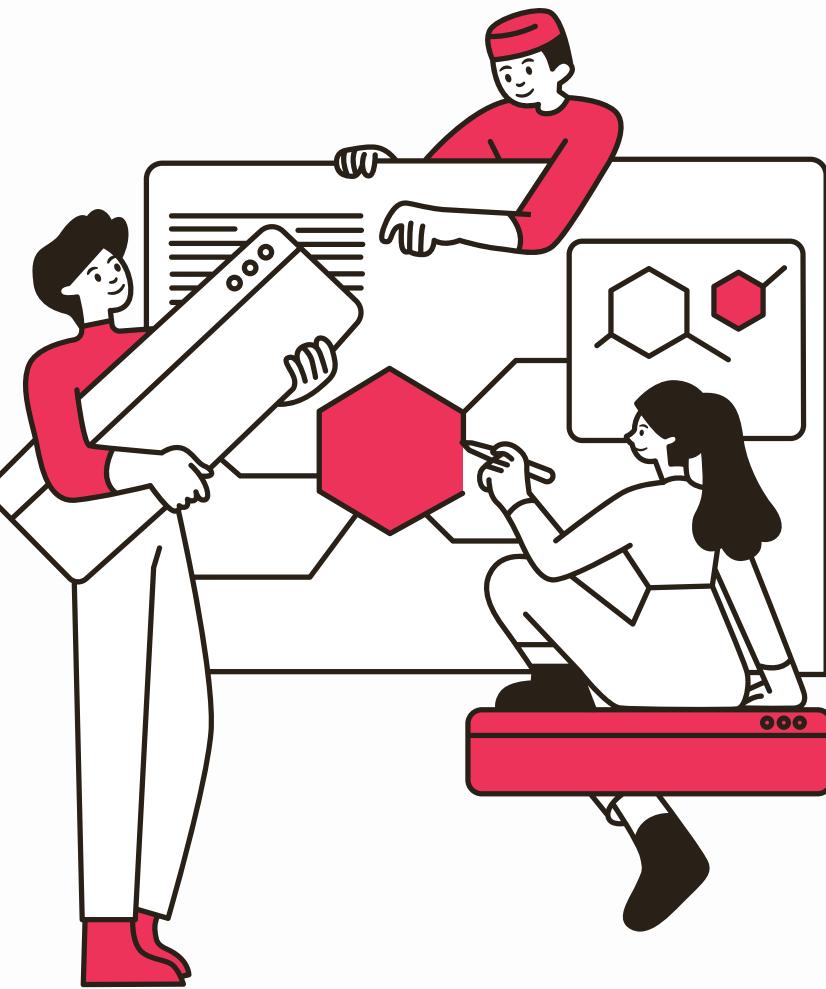


Year  
2024-25

Presenter  
Gruppo 09

# Proposta di Progetto

Design proposti per lo sviluppo di una Rubrica Telefonica



Date  
17/12/2024

# Executive Summary

La Rubrica Telefonica è un'applicazione software progettata per semplificare la gestione e l'organizzazione dei contatti personali e professionali. Grazie a un'interfaccia intuitiva, la rubrica offre una soluzione moderna e accessibile per mantenere i contatti sempre aggiornati.



# Product Strategy

01

## Content Design

La Rubrica Telefonica offre un'interfaccia pulita e intuitiva, con un design smart e professionale. Ogni elemento è progettato per essere accessibile e facilmente navigabile.

02

## Visual Design

L'applicazione adotta le linee guida di Material Design per garantire un sistema visivo coeso e reattivo su ogni dispositivo (Windows, macOS, Linux).

03

## Interaction Design

La navigazione è fluida e intuitiva: ogni funzione è accessibile in pochi clic. Gli utenti possono aggiungere, modificare, eliminare e cercare contatti con facilità.

# Contenuto

03

CONTENT



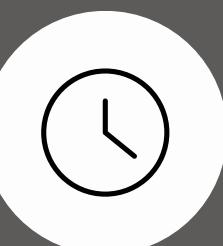
Requisiti e Casi d'uso



Diagrammi

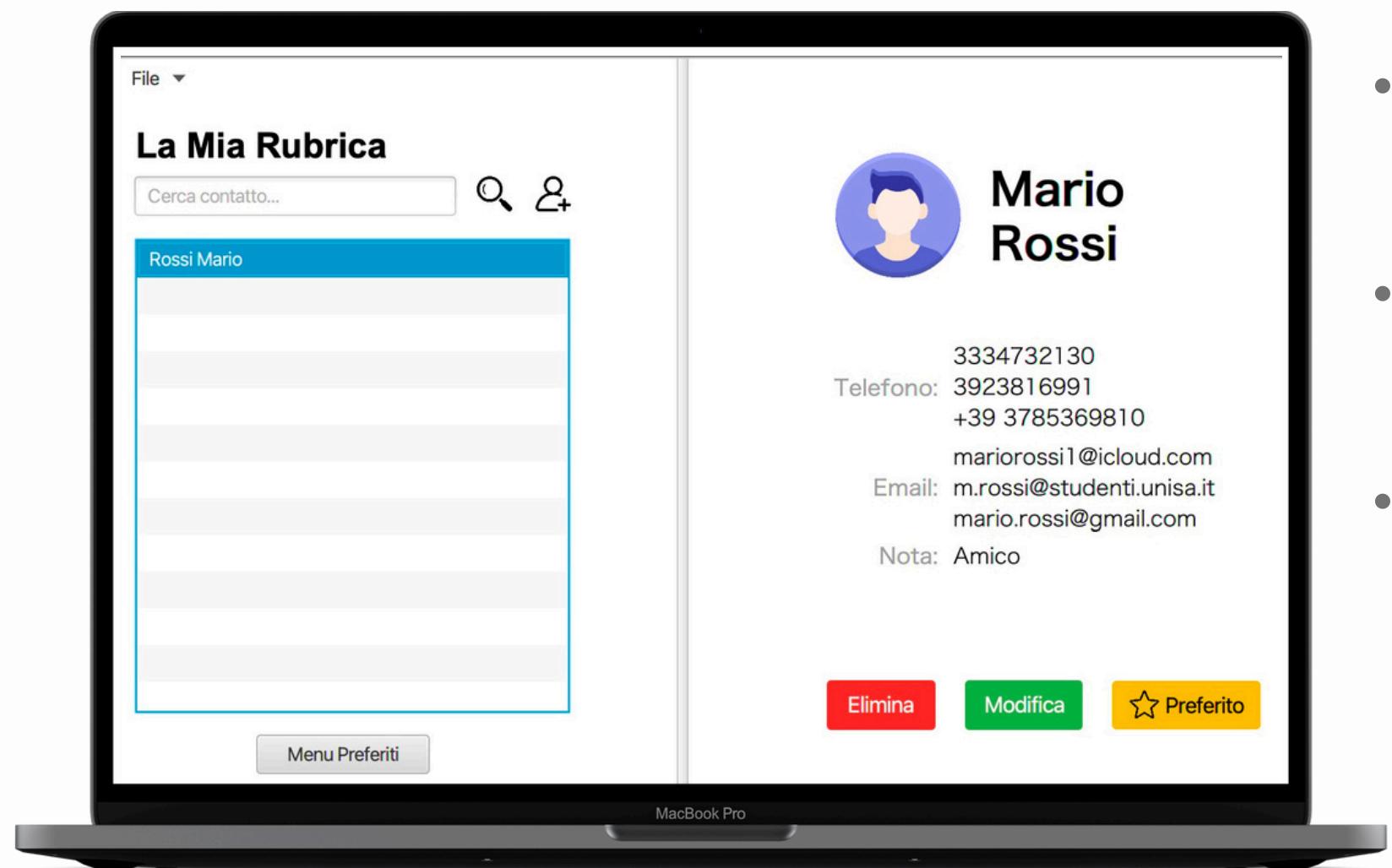


Cambiamenti in Corso d'Opera



Demo del Programma Funzionante

# Introduzione



- Proposito: Sviluppare una rubrica telefonica digitale che faciliti la gestione dei contatti in modo semplice ed efficiente.
- Obiettivo: Creare un'applicazione multi-piattaforma con funzionalità essenziali come ricerca rapida, organizzazione in gruppi e gestione dei contatti.
- Utenti Target: Professionisti, studenti e utenti domestici in cerca di una soluzione affidabile per la gestione dei contatti.

05

REQUIREMENTS

# Requisiti

01

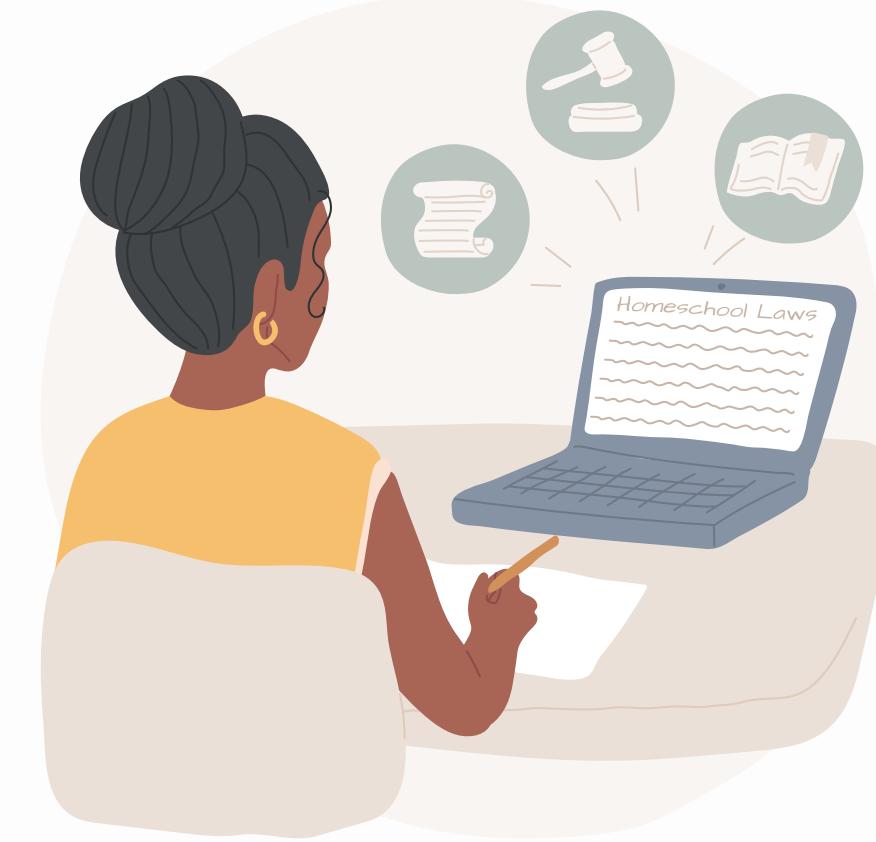
## Requisiti Funzionali

- Aggiungere, modificare, eliminare contatti.
- Funzione di ricerca avanzata.
- Importazione ed Esportazione della Rubrica.

02

## Requisiti Non Funzionali

- Tempo di risposta inferiore a 2 secondi per operazioni di ricerca.
- Interfaccia intuitiva con design moderno.
- Compatibilità con i principali sistemi operativi.



# Casi d'Uso

Aggiunta di un Contatto

Ricerca di un Contatto

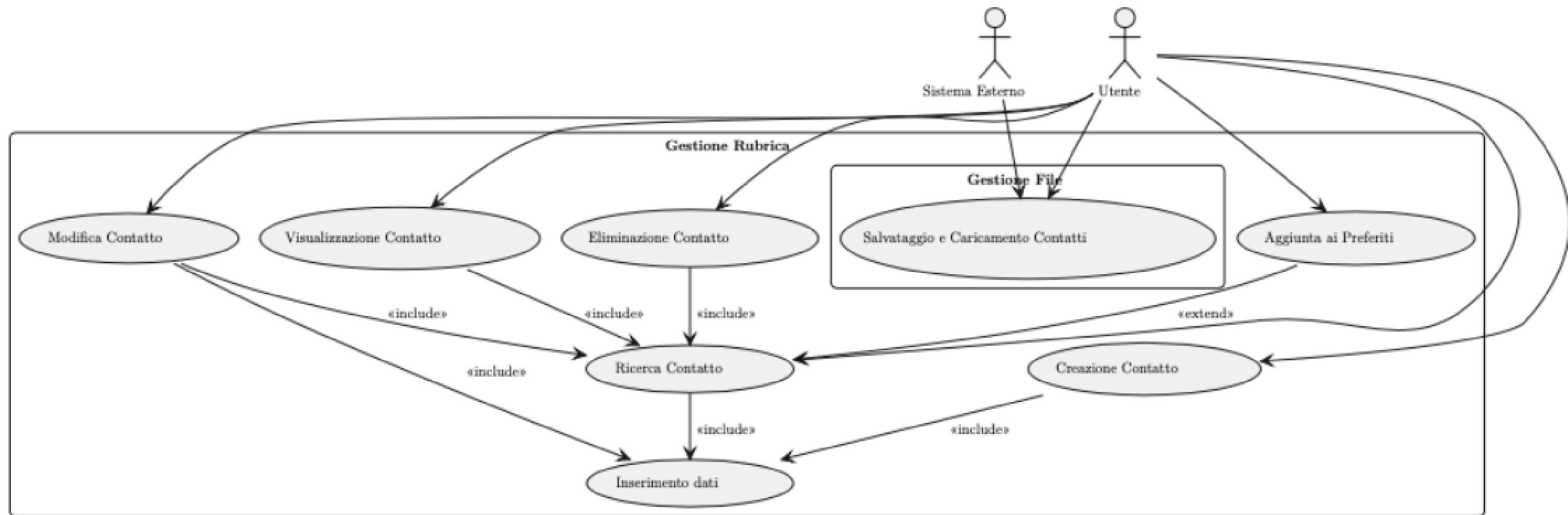
Eliminazione di un Contatto

Aggiunta del Contatto ai Preferiti

Modifica di un Contatto

Salvataggio Caricamento Dati

# Casi d'Uso



# Design Architetturale

07

01

## Model

Gestisce i dati e la logica di business.

Classe Model	Descrizione
Contatto	Rappresenta un singolo contatto con attributi come nome, cognome, telefono, email e note.
Rubrica	Gestisce la lista principale dei contatti e la lista dei preferiti.
SalvaCaricaRubrica	Si occupa della persistenza dei dati della rubrica su file JSON o CSV.
SalvaCaricaPreferiti	Si occupa della persistenza dei contatti preferiti su file JSON.
ContattoValidator	Valida i campi inseriti (nome, email, telefono) e rileva eventuali duplicati.

02

## View

Rappresenta l'interfaccia utente.

View (FXML)	Descrizione	Controller Associato
InterfacciaUtente.fxml	Gestione rubrica principale	InterfacciaUtenteController.java
MenuPreferiti.fxml	Gestione dei contatti preferiti	MenuPreferitiController.java
InterfacciaAggiungiModifica.fxml	Aggiunta e modifica contatti	InterfacciaAggiungiModificaController.java

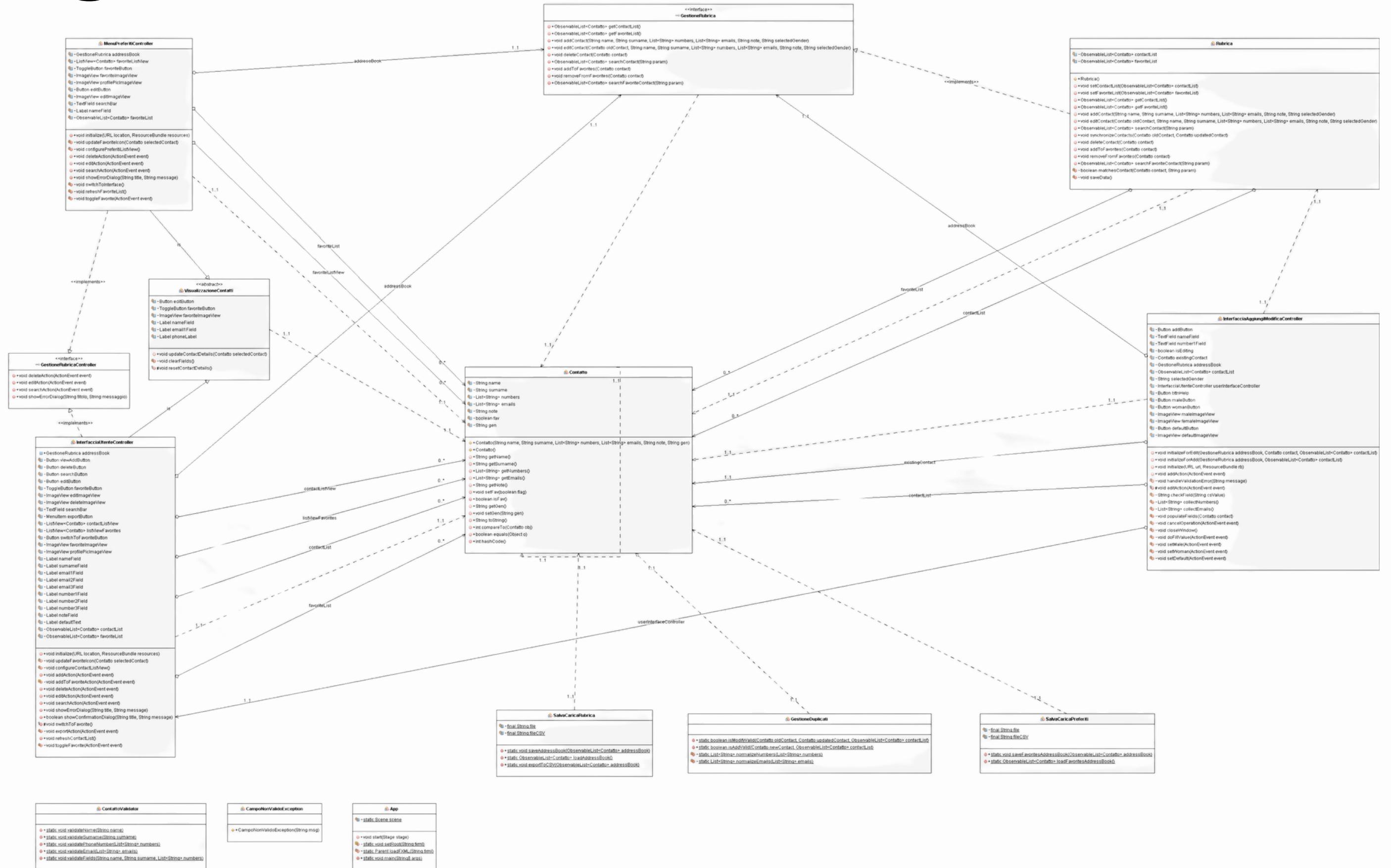
03

## Controller

Gestisce gli input dell'utente e aggiorna il modello.

Controller	Descrizione	View Associata
InterfacciaUtenteController.java	Gestisce l'interfaccia principale della rubrica, inclusa la visualizzazione dati.	InterfacciaUtente.fxml
MenuPreferitiController.java	Gestisce la lista dei contatti preferiti, inclusi aggiornamenti e rimozioni.	MenuPreferiti.fxml
InterfacciaAggiungiModificaController.java	Aggiunta e modifica contatti	InterfacciaAggiungiModifica.fxml

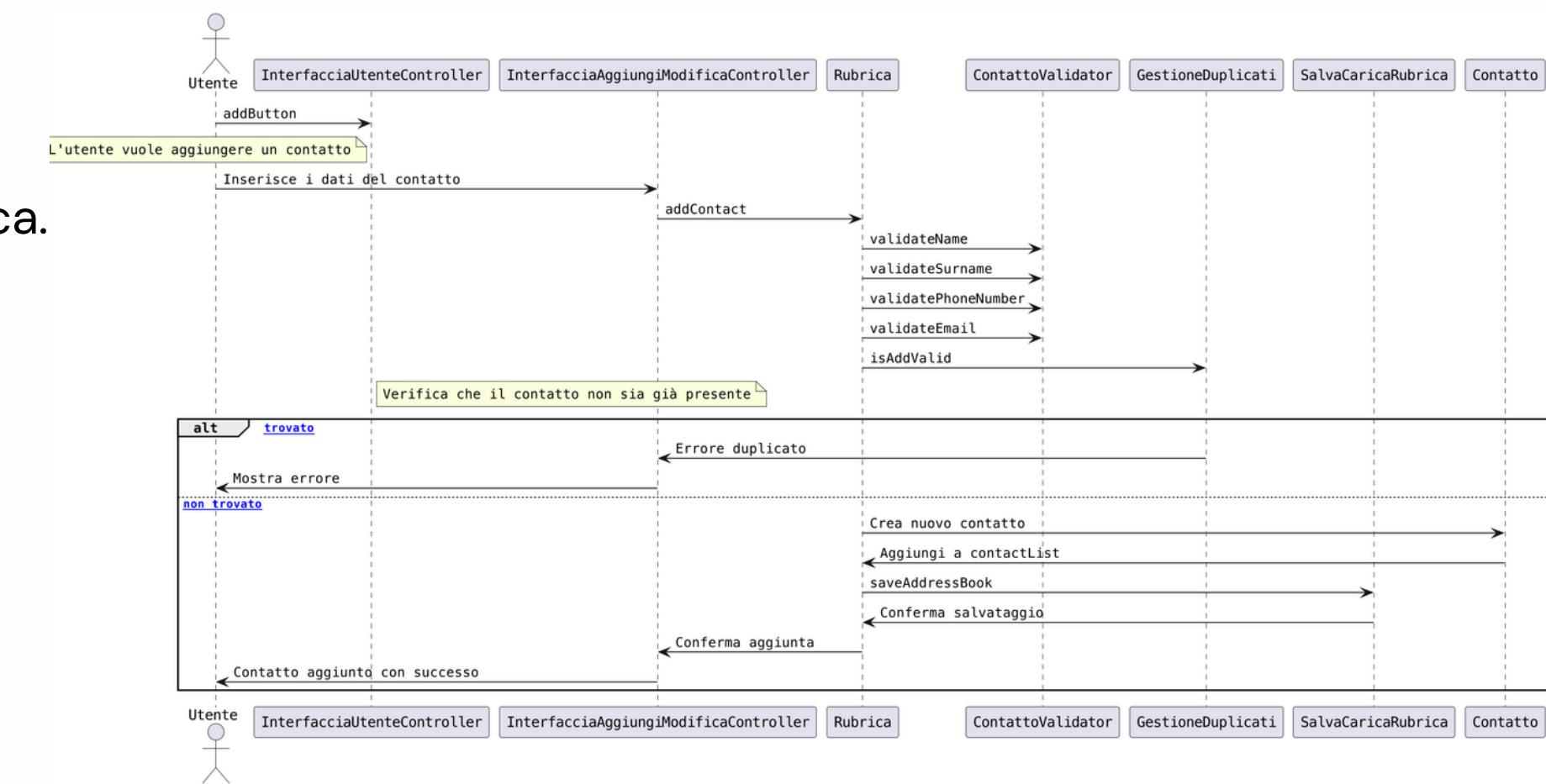
# Diagramma delle Classi



# Diagrammi di Sequenza

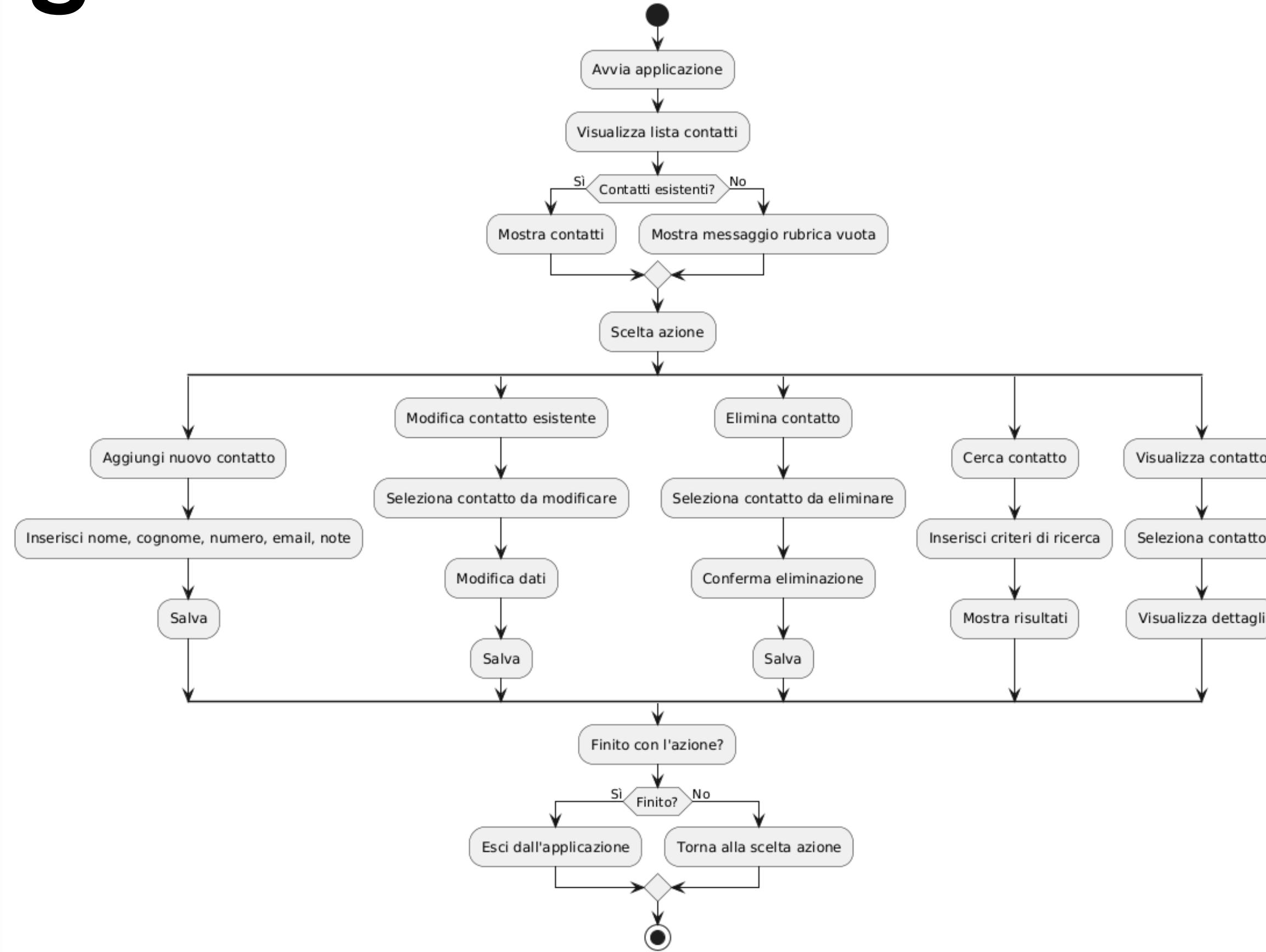
## Caso d'Uso: Aggiungere un Contatto

- Aggiungere, modificare, eliminare contatti.
- L'utente inserisce i dati.
- Il Controller verifica la validità.
- Il Model salva i dati.
- La View aggiorna l'interfaccia con il nuovo contatto.
- Importazione ed Esportazione della Rubrica.



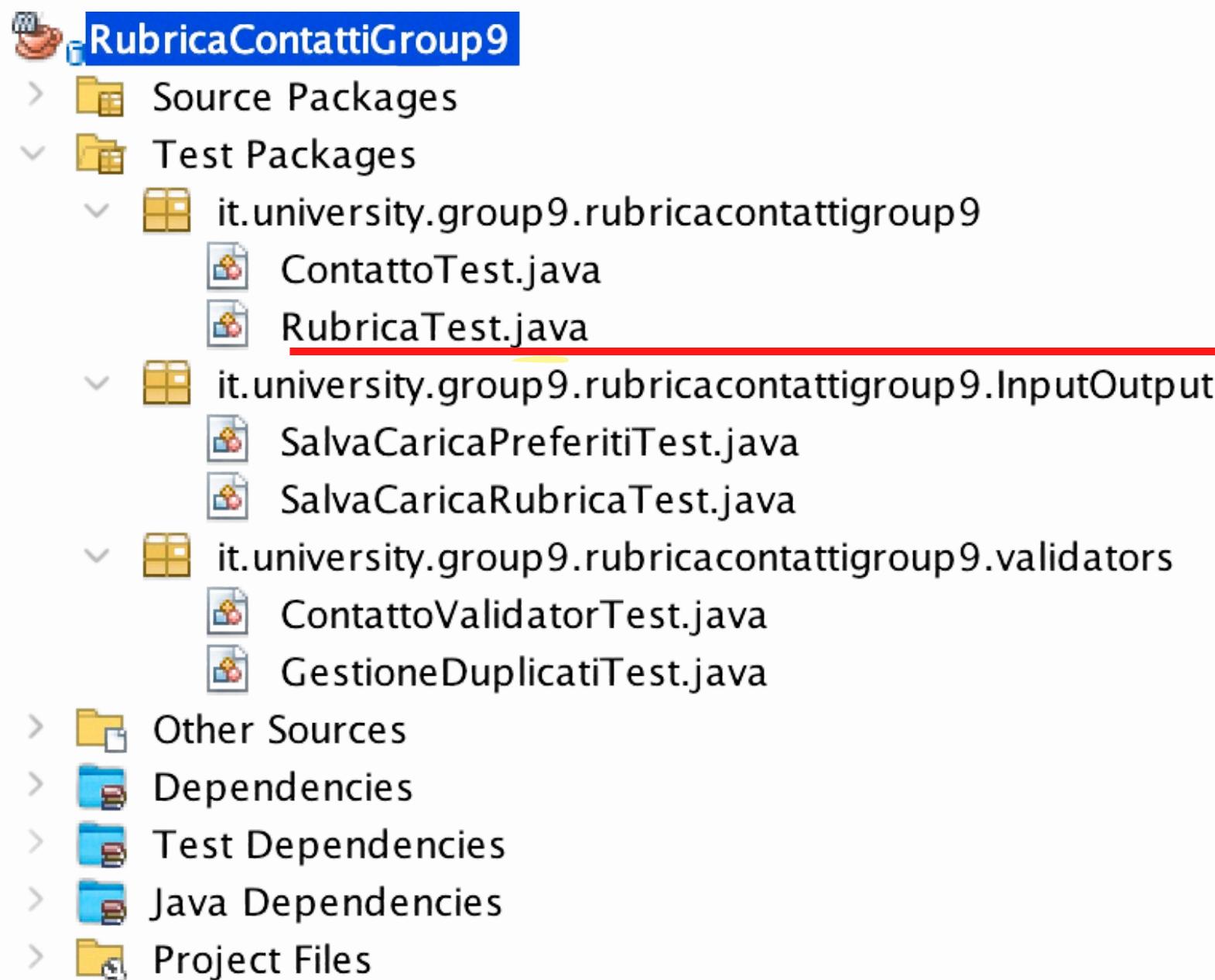
# Diagramma delle Attività

10



# Casi di Test

11

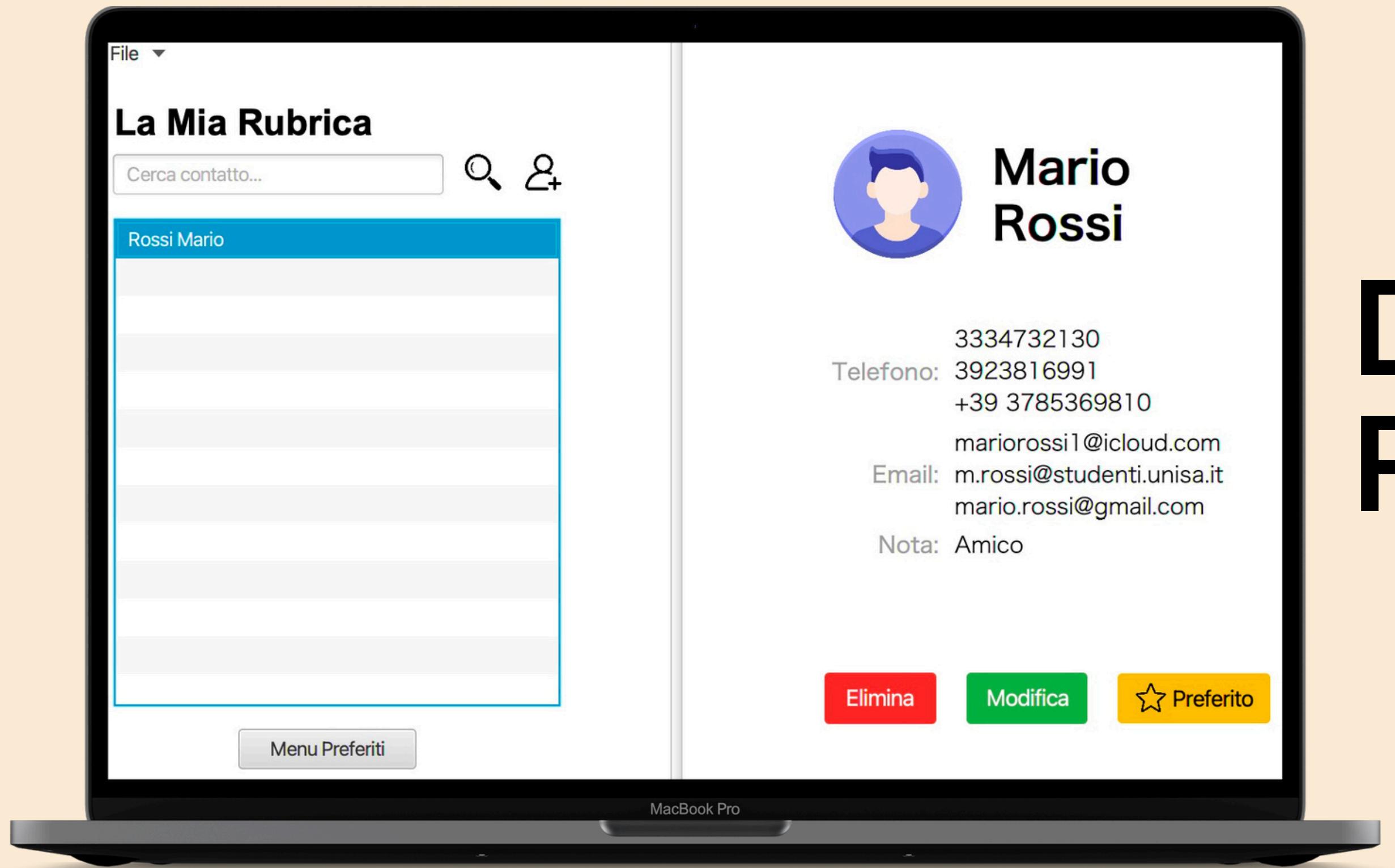


Test	Rubrica
TestItems	testEditContact_ValidEdit_ShouldUpdateContact
Input	<pre> rubrica = new Rubrica(); rubrica.setContactList(FXCollections.observableArrayList()); rubrica.setFavoriteList(FXCollections.observableArrayList()); Contatto oldContact = new Contatto("Dati"); rubrica.getContactList().add(oldContact); String newName = ""; String newSurname = ""; List&lt;String&gt; newNumbers = Arrays.asList(""); List&lt;String&gt; newEmails = Arrays.asList(""); String newNote = ""; String newGen = ""; rubrica.editContact(oldContact, newName, newSurname, newNumbers, newEmails, newNote, "newGen"); </pre>
Oracle	<pre> contactList.size() == 1; updatedContact.getName() == newName; updatedContact.getSurname() == newSurname; updatedContact.getNumbers() == newNumbers; updatedContact.getEmails() == newEmails; updatedContact.getNote() == newNote; updatedContact.getNote() == newGen; </pre>

# Analisi di Coesione e Accoppiamento

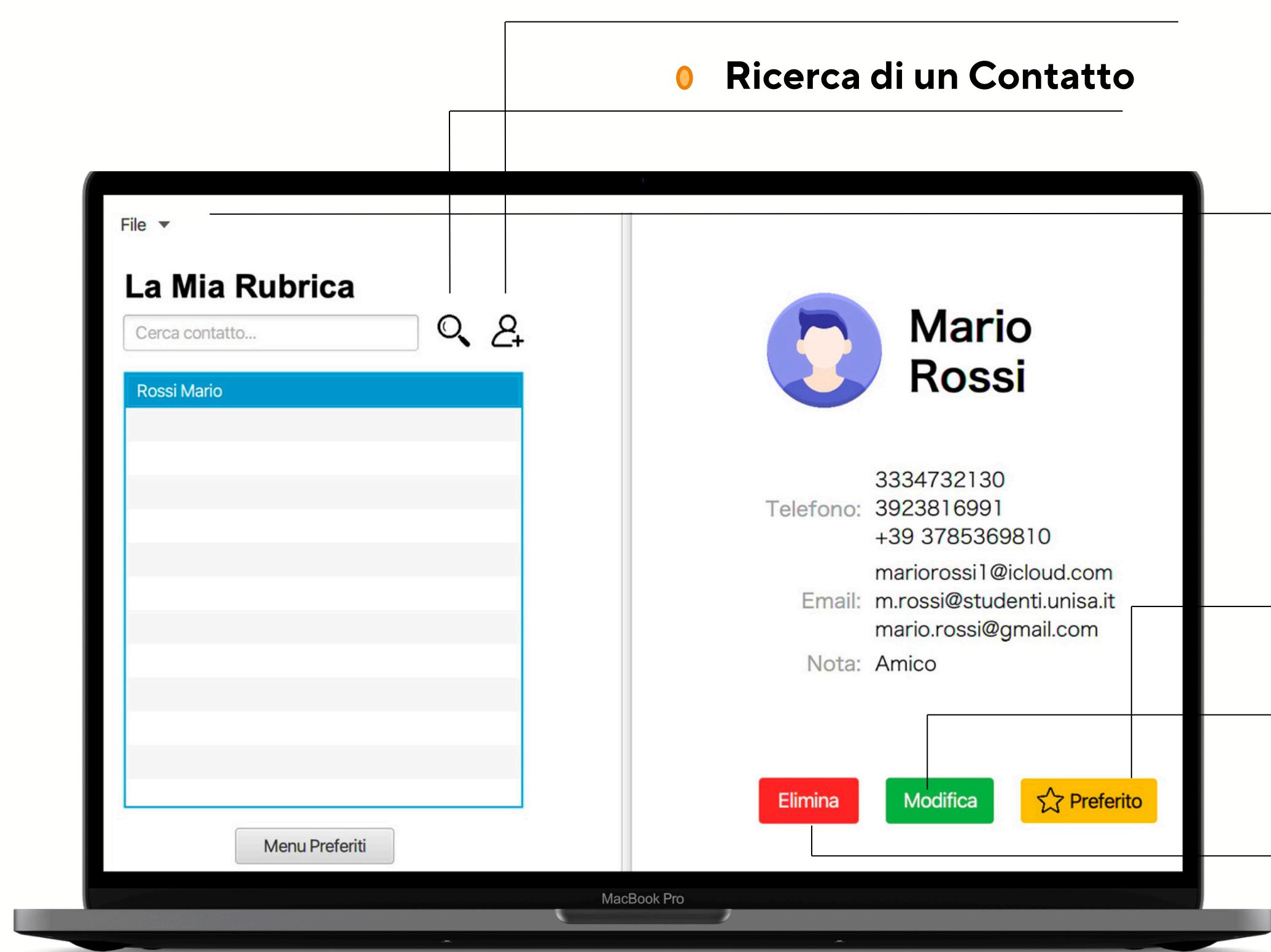
Modulo	Livello di Coesione	Descrizione
Contatto	Funzionale	Rappresenta l'entità "Contatto" con tutte le sue proprietà (nome, cognome, numeri, email, note, genere).
Rubrica	Funzionale	Implementa tutte le operazioni relative alla gestione della rubrica (aggiunta, modifica, eliminazione, ricerca, gestione dei preferiti).
InterfacciaUtenteController	Funzionale	Gestisce tutte le interazioni con l'utente attraverso l'interfaccia grafica.
MenuPreferitiController	Funzionale	Le funzionalità si concentrano sulla gestione dei contatti preferiti.
InterfacciaAggiungiModificaController	Procedurale	Gestisce il flusso di lavoro per aggiungere o modificare un contatto. Le operazioni includono la raccolta dei dati e il passaggio del contatto aggiornato alla rubrica.
SalvaCaricaRubrica	Funzionale	Lo scopo principale è gestire il salvataggio e il caricamento di dati (in formato JSON o CSV).
SalvaCaricaPreferiti	Funzionale	Simile a SalvaCaricaRubrica, ma specifico per i contatti preferiti.
GestioneDuplicati	Logica	Le operazioni effettuate sono simili, ma in contesti diversi.
CampoNonValidoException	Funzionale	Segnala errori relativi alla validazione dei campi, con metodi dedicati a costruire e restituire messaggi di errore.
ContattoValidator	Logica	Controlla la validità dei campi (nome, cognome, numeri di telefono, email) per garantire l'integrità dei dati.

Modulo	Livello di Accoppiamento	Descrizione
Contatto	Per Dati	Si è scelto un accoppiamento minimo per trasferire solo i dati necessari, garantendo che la classe resti focalizzata sulla rappresentazione del modello.
Rubrica	Per Timbro, Per Dati	Le sue dipendenze sono coerenti con il suo ruolo come gestore centrale dei dati. Tuttavia, alcuni metodi come <code>searchContact</code> dipendono da parametri esterni.
InterfacciaUtenteController	Per Timbro, Per Dati	L'accoppiamento con l'interfaccia GestioneRubrica è necessario per trasferire dati complessi, come le liste osservabili (rubriche) o dei numeri, tra l'interfaccia grafica e il modello, garantendo sincronizzazione dinamica e aggiornamenti in tempo reale.
MenuPreferitiController	Per Timbro, Per Dati	Le liste osservabili sono passate per facilitare la gestione dinamica dei dati preferiti, minimizzando la complessità di sincronizzazione con la rubrica principale.
InterfacciaAggiungiModificaController	Per Dati	Si è scelto un accoppiamento per dati essenziali per garantire una gestione diretta delle operazioni di inserimento e modifica.
SalvaCaricaRubrica	Per Timbro	L'accoppiamento per timbro è necessario per scambiare in modo sicuro e consistente liste osservabili tra la rubrica dei preferiti e i file JSON, sfruttando la libreria Jackson.
SalvaCaricaPreferiti	Per Timbro	L'accoppiamento è il medesimo di SalvaCaricaRubrica.
GestioneDuplicati	Per Timbro, Per Dati	L'accoppiamento consente di verificare duplicati direttamente sulle liste osservabili.
CampoNonValidoException	Per Dati	Un accoppiamento per dati è sufficiente per segnalare errori specifici.
ContattoValidator	Per Dati	L'accoppiamento consente un controllo rigoroso della validità dei dati, segnalando errori attraverso eccezioni, mantenendo la separazione dei compiti.



# DEMO del Prodotto

# Features



## ● Aggiunta di un Contatto

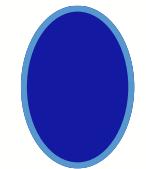
## ● Ricerca di un Contatto

## ● Esportazione della Rubrica in formato CSV

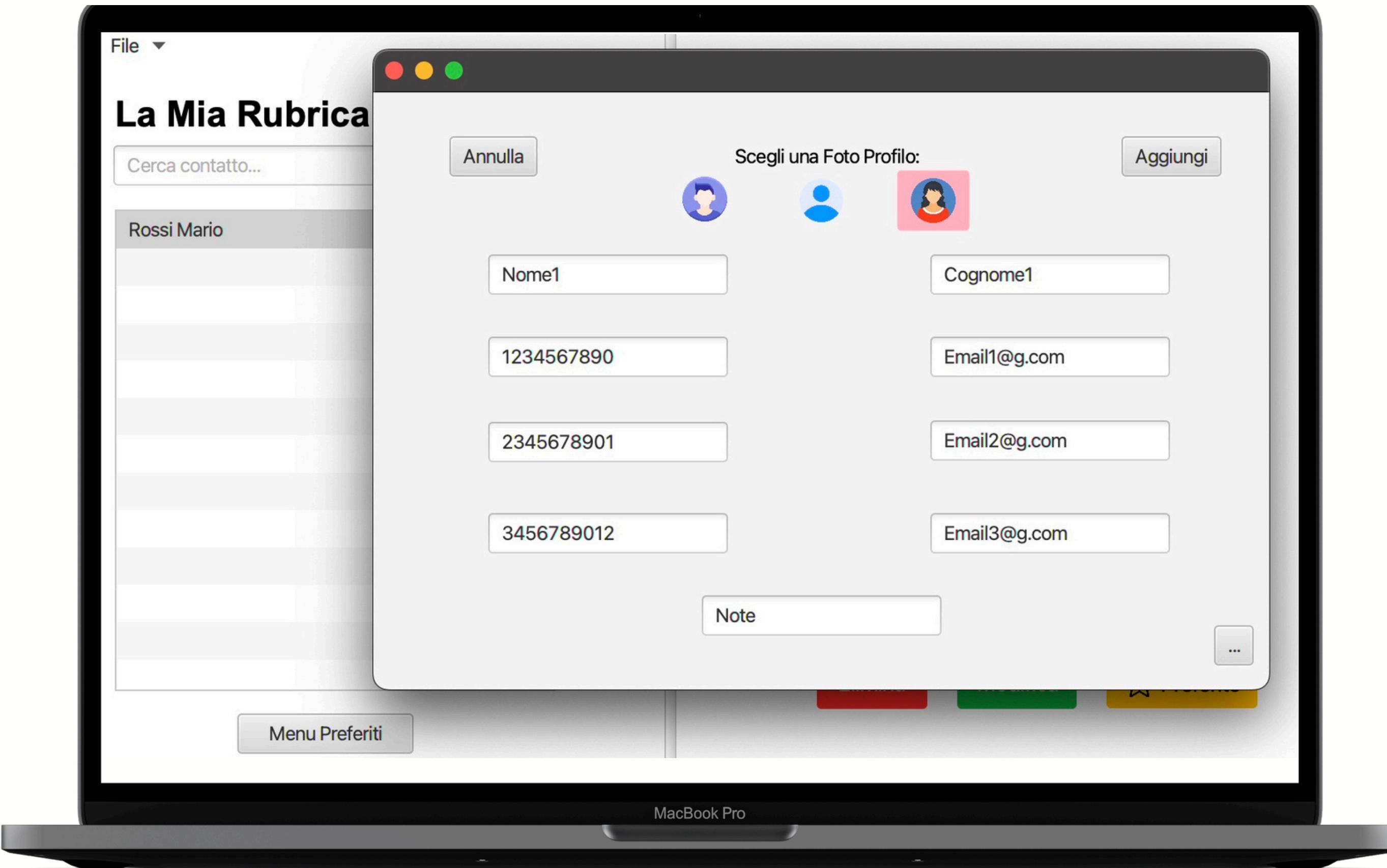
## ● Aggiunta di un Contatto ai Preferiti

## ● Modifica di un Contatto

## ● Eliminazione di un Contatto

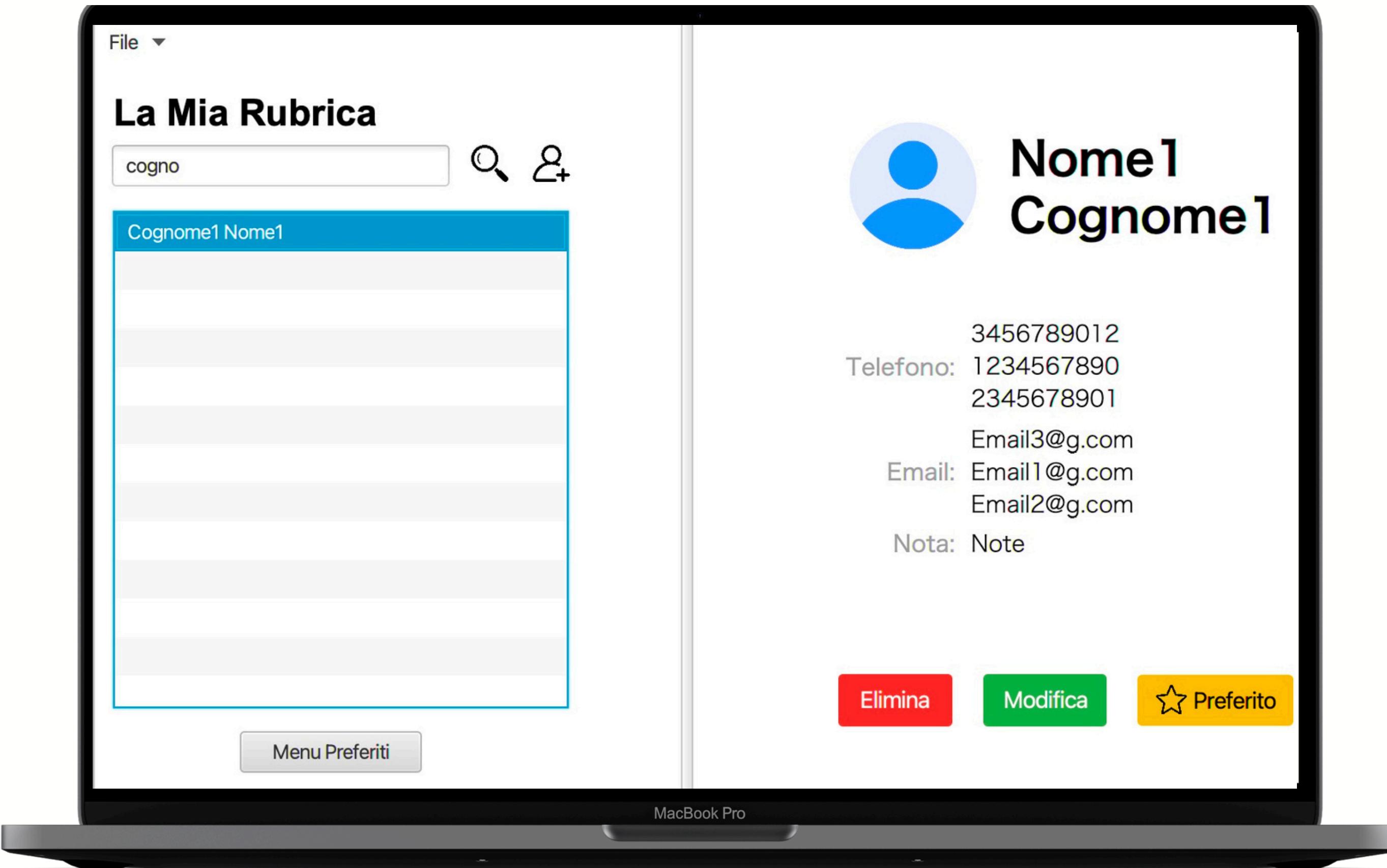


# Aggiunta di un Contatto



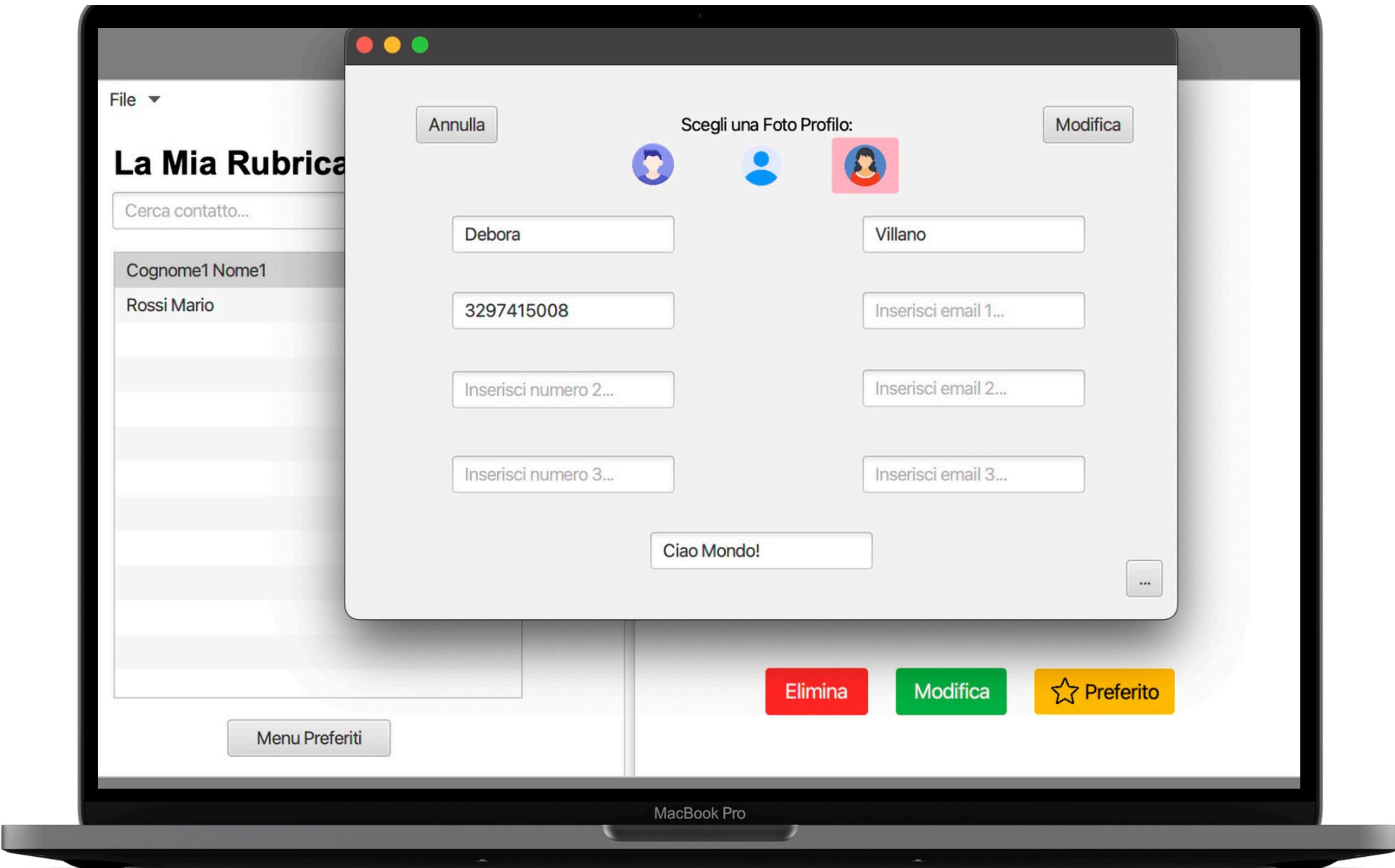
Features

# Ricerca di un Contatto



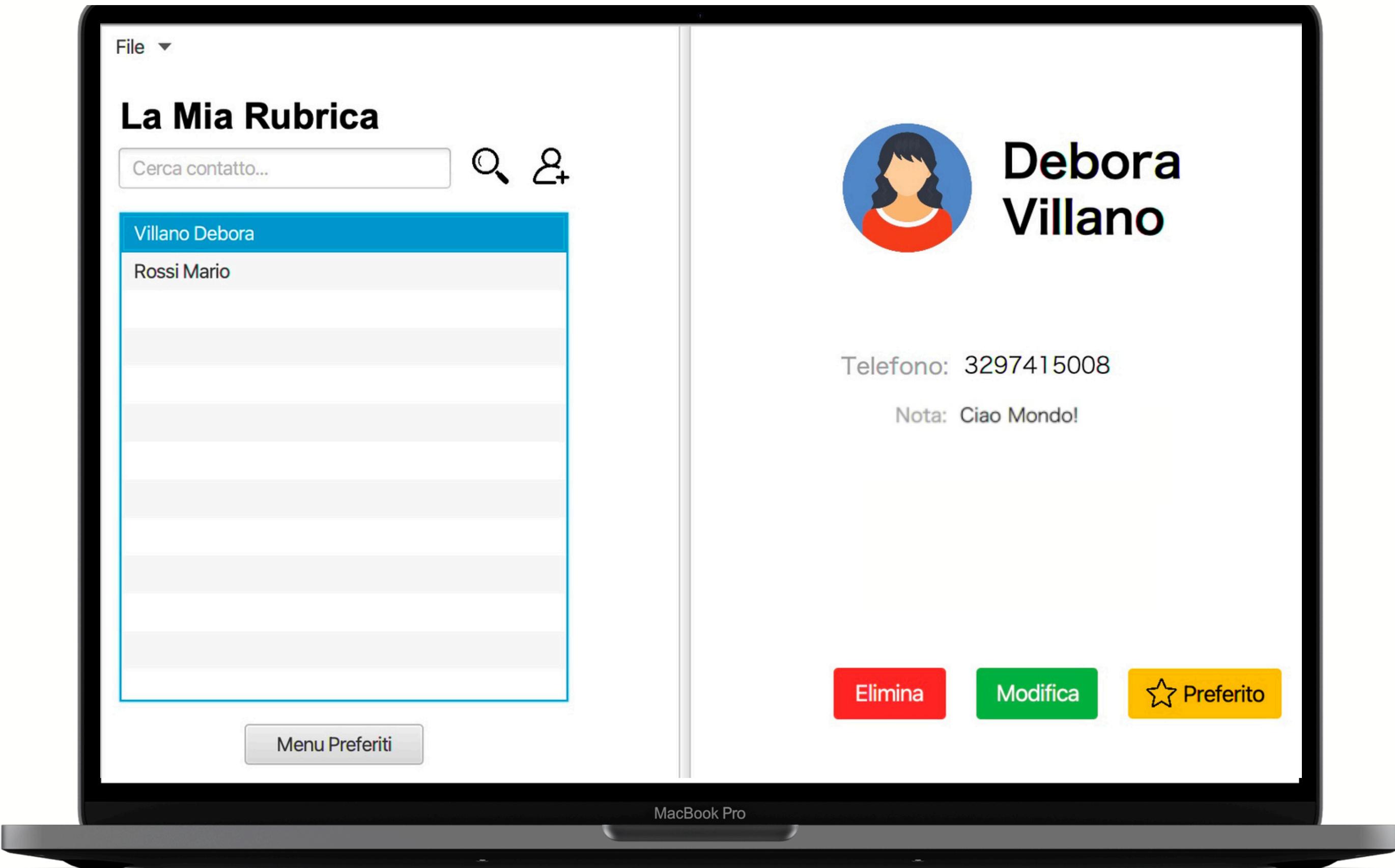
Features

# Modifica di un Contatto



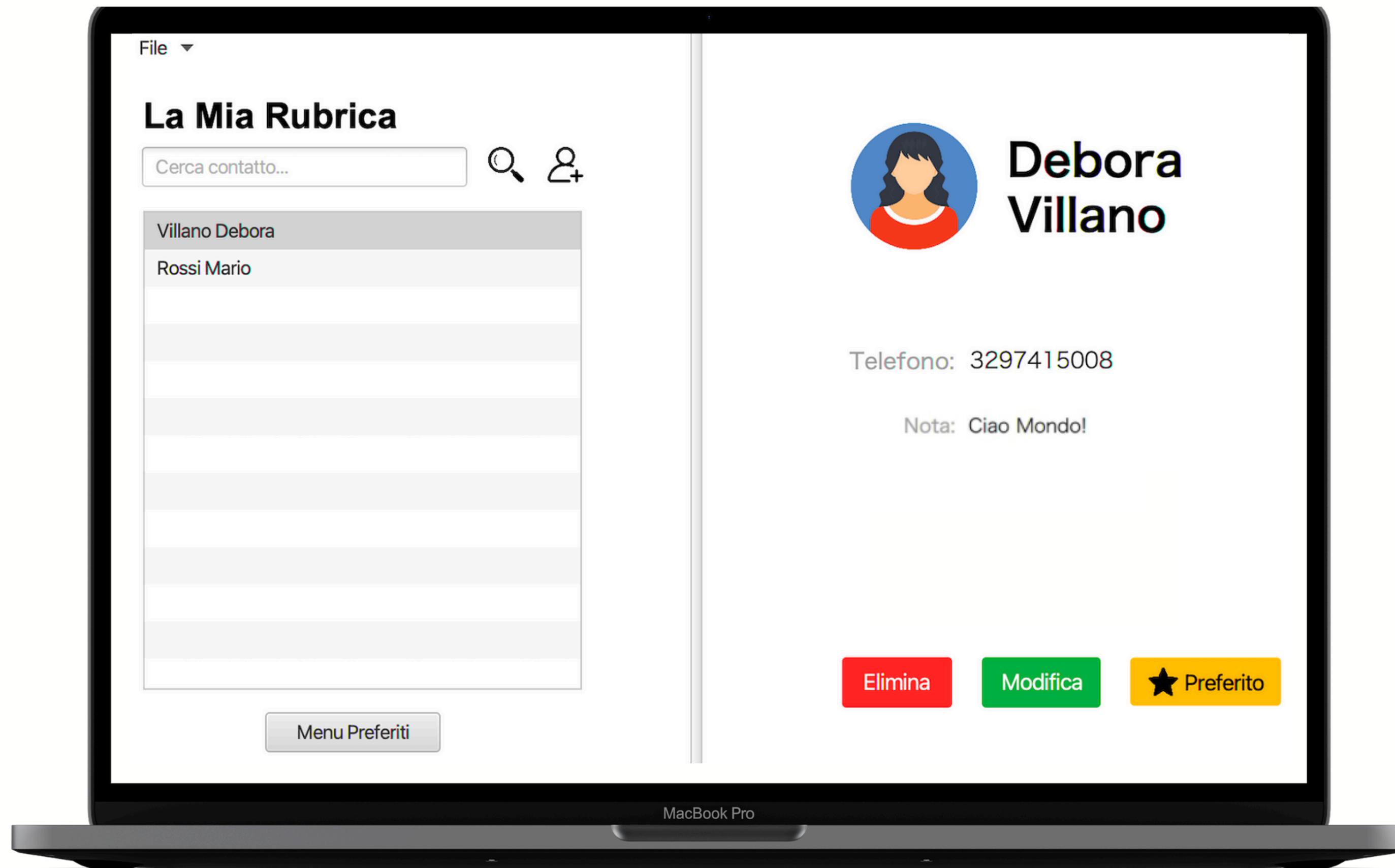
Features

# Modifica di un Contatto



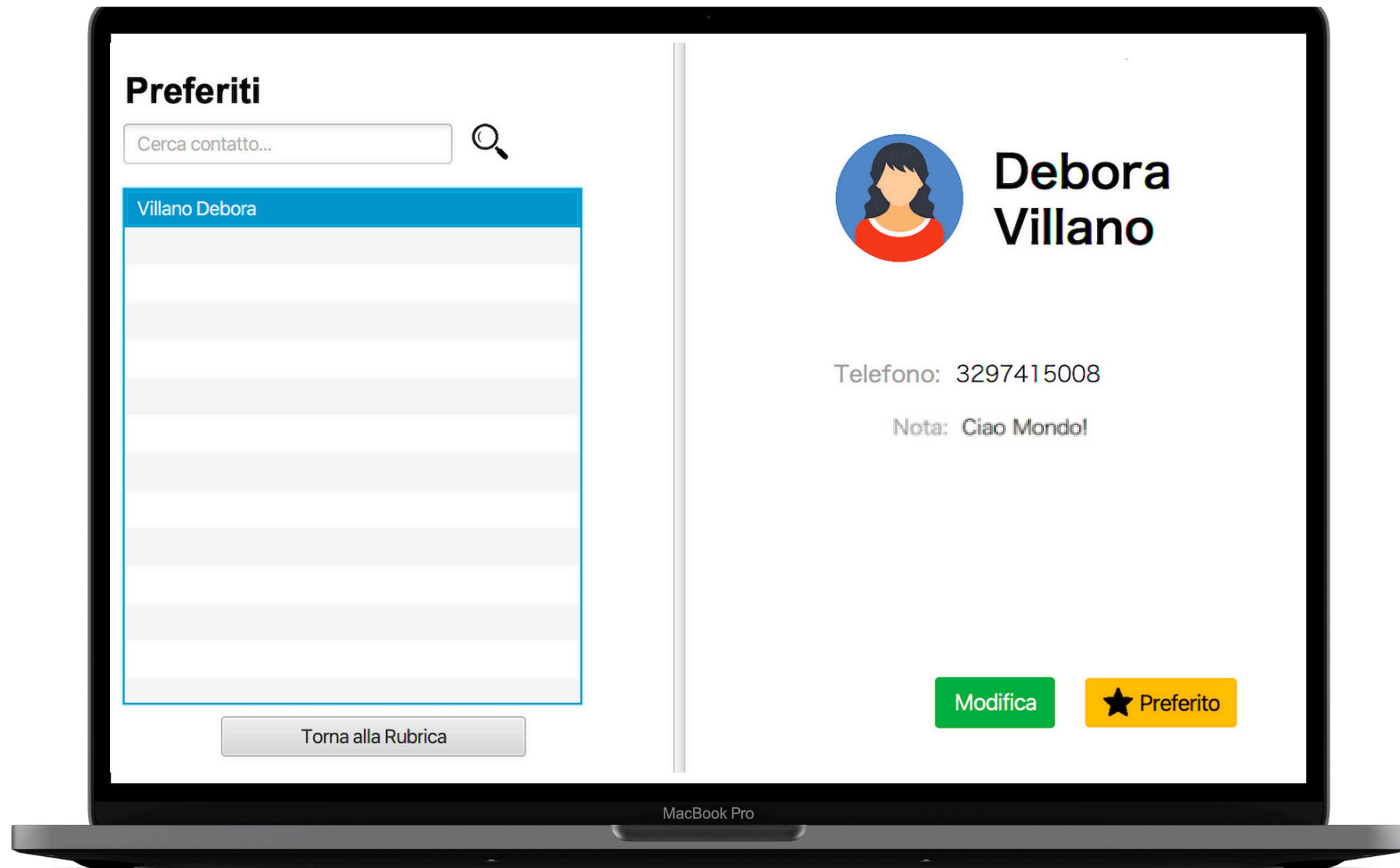
Features

# ● Aggiunta di un Contatto ai Preferiti



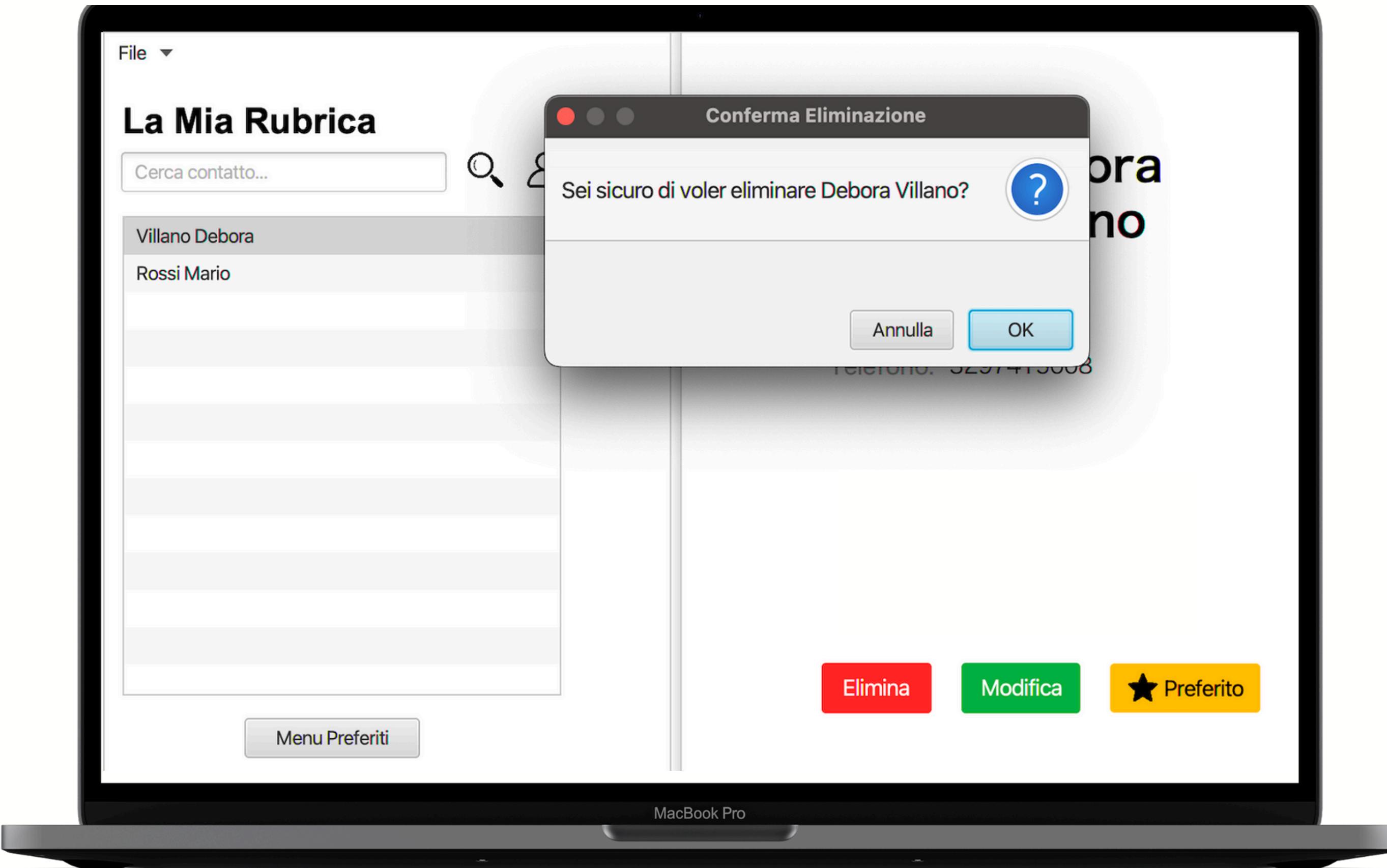
Features

# ● Aggiunta di un Contatto ai Preferiti



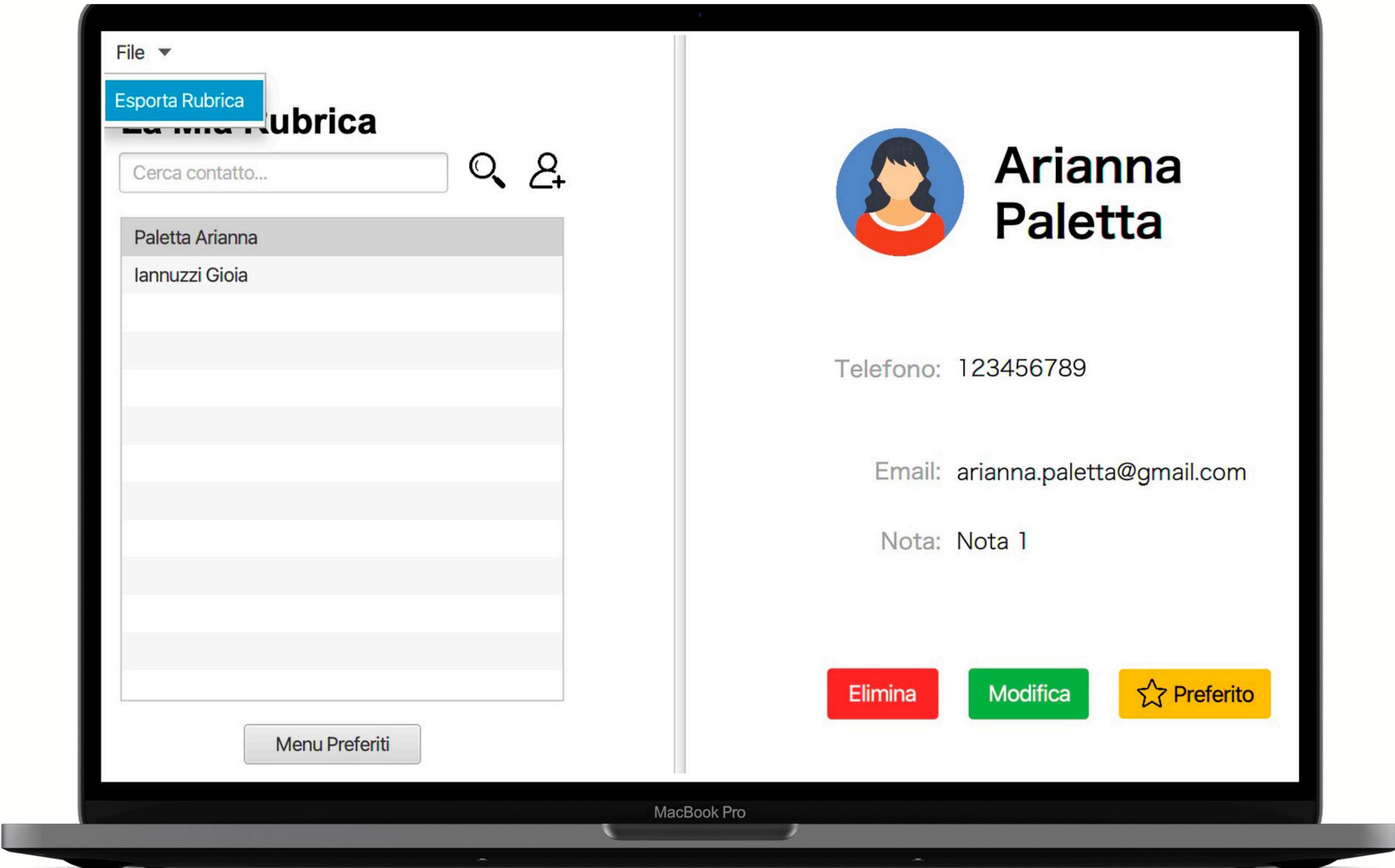
Features

# ● Eliminazione di un Contatto



Features

# ● Esportazione della Rubrica in formato CSV



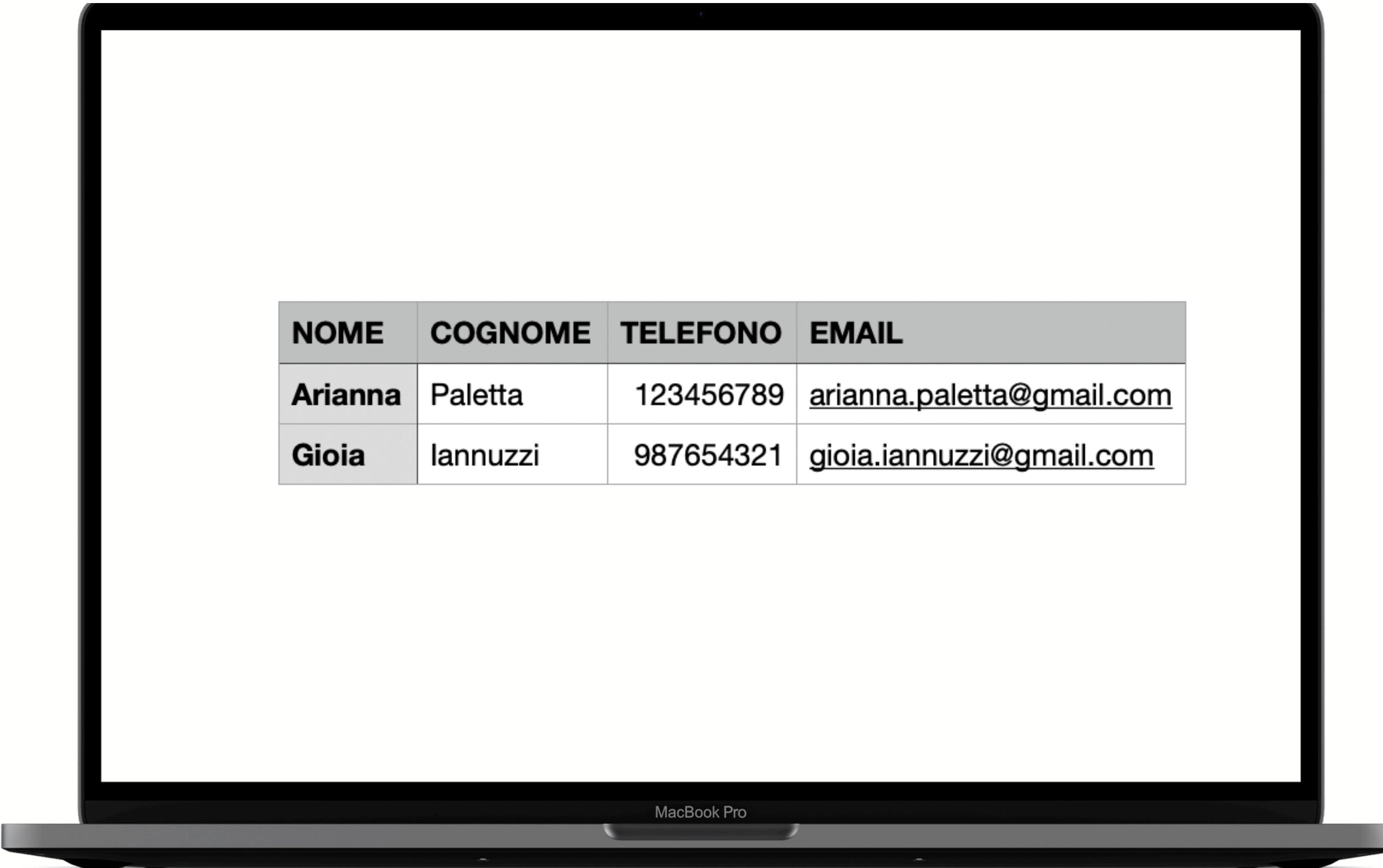
Features

# ● Esportazione della Rubrica in formato CSV



Features

# ● Esportazione della Rubrica in formato CSV



MacBook Pro

Features

# Conclusioni

## Risultati Raggiunti

- Sistema funzionante con architettura robusta.
- Funzionalità principali implementate con successo.

## Sviluppi Futuri

- Integrazione con servizi cloud per backup automatici.
- Sincronizzazione multi-dispositivo.

## Contatti

- Gruppo 9
- 0612707416 – Gioia Iannuzzi
- 0612708273 – Luigi Montonetti
- 0612707862 – Arianna Paletta
- 0612708294 – Debora Villano